**Звіт**

**Лабораторна работа 10. Обробка параметризованих контейнерів**

**Мета роботи**: Розширення функціональності параметризованих класів.

**ВИМОГИ**

Використовуючи програму рішення завдання лабораторної роботи №9:

1. Розробити параметризовані методи (Generic Methods) для обробки колекцій об'єктів згідно прикладної задачі.
2. Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах.

* Автоматичний режим виконання програми задається параметром командного рядка **-auto**. Наприклад, java ClassName -auto.
* В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.

1. Забороняється використання алгоритмів з Java Collections Framework.
   1. **Розробник**: Момот Роман Євгенійович, КІТ119а, варіант №14.
2. **ОПИС ПРОГРАМИ**
   1. **Засоби ООП**: клас, метод класу, поле класу.
   2. **Ієрархія та структура класів:** один публічний клас Main, публічний клас Event, у полях якого є час початку події, тривалість, адреса події, імена людей, опис події, гетери, сетери, конструктор класу та метод виведення даних класу. Також є клас Node, який виконує роль покажчика на елемент і клас MyContainer, який містить покажчик на головний елемент та методи обробки масиву елементів.
   3. **Важливі фрагменти програми:**

public class Main {

public static void main(String[] args) {

MyContainer<Event> arr = new MyContainer<Event>();

for(String str: args)

{

if(str.equals("-a") || str.equals("-auto")) {

arr = auto(arr);

return;

}

else if(str.equals("-d") || str.equals("-dialog")) {

arr = menu(arr);

return;

}

}

arr = menu(arr);

}

private static MyContainer<Event> auto(MyContainer<Event> arr) {

System.out.println("\nSize of container: " + arr.getSize());

System.out.println("Adding elements...");

ArrayList<String> people = new ArrayList<String>();

people.add("John");

people.add("Bill");

people.add("Івасик");

Event event = new Event(new GregorianCalendar(28,4,2002), 120, "ул. Революции",

people, "Pest party ever");

arr.add(event);

people = new ArrayList<String>();

people.add("Roman");

people.add("Dmitriy");

event = new Event(new GregorianCalendar(15,10,2020), 30, "пр. Тракторостроителей",

people, "So boring");

arr.add(event);

System.out.println("Size of container: " + arr.getSize());

System.out.println("\nOutputing data with toArray:");

Object[] tempArr = arr.toArray();

for (int i = 0; i < tempArr.length; i++) {

System.out.println(i+1 + ")");

((Event)tempArr[i]).outputData();

System.out.println( );

}

System.out.println("Is container empty?");

System.out.println(arr.isEmpty());

return arr;

}

private static MyContainer<Event> menu(MyContainer<Event> arr) {

Scanner scan = new Scanner(System.in);

boolean stop = false;

int choise, choise2;

ArrayList<String> people = new ArrayList<String>();

people.add("John");

people.add("Bill");

people.add("Івасик");

Event evToCompare = new Event(new GregorianCalendar(2002,3,28), 120, "ул. Революции",

people, "Pest party ever");

arr.add(evToCompare);

do {

System.out.println("What to do?");

System.out.println("1. Output data");

System.out.println("2. Add element");

System.out.println("3. Delete element");

System.out.println("4. Find element");

System.out.println("5. Serialization");

System.out.println("6. Deserialization");

System.out.println("7. Sort data");

System.out.println("8. Terminate program");

System.out.println("=====================");

System.out.print("Your choise: ");

choise = scan.nextInt();

switch(choise) {

case 1:

System.out.println("\nChoose the output method");

System.out.println("1. Using foreach");

System.out.println("2. Using toArray");

System.out.println("3. Return");

System.out.println("================");

System.out.print("Your choise: ");

choise2 = scan.nextInt();

System.out.println( );

switch(choise2) {

case 1:

if(arr.getSize() > 0){

for(var i : arr) {

i.outputData();

System.out.println( );

}

System.out.println( );

}

else {

System.out.println("Array is empty.\n");

}

break;

case 2:

if(arr.getSize() > 0) {

Object[] tempArr = arr.toArray();

for (int i = 0; i < tempArr.length; i++) {

System.out.println(i + ")");

((Event)tempArr[i]).outputData();

}

}

else {

System.out.println("Array is empty.");

}

break;

case 3:

break;

default:

System.out.println("You've entered the wrong number");

break;

}

break;

case 2:

Event newEvent = inputNewEvent();

arr.add(newEvent);

break;

case 3:

if(arr.getSize() > 0) {

System.out.print("Enter the index of element: ");

choise = scan.nextInt();

arr.delete(choise);

} else {

System.out.println("Array is empty.");

}

break;

case 4:

if(arr.isEmpty()) {

System.out.println("Array is empty.");

} else {

System.out.println("Array isn't empty.");

}

break;

case 5:

System.out.println("\nChoose the method");

System.out.println("1. Standard serialization");

System.out.println("2. XML serialization");

System.out.println("3. Return");

System.out.println("=========================");

System.out.print("Your choise: ");

choise2 = scan.nextInt();

switch(choise2) {

case 1:

scan.nextLine();

System.out.print("Enter the name of file: ");

String filename = scan.nextLine();

if (filename.indexOf(".ser") == -1) {

filename += ".ser";

}

try(ObjectOutputStream oos = new ObjectOutputStream(new BufferedOutputStream(new FileOutputStream(filename)))){

oos.writeObject(arr);

System.out.println("Serialization successful.");

}catch(Exception ex){

System.out.println(ex.getMessage());

ex.printStackTrace();

}

break;

case 2:

scan.nextLine();

System.out.print("Enter the name of file: ");

filename = scan.nextLine();

if (filename.indexOf(".xml") == -1) {

filename += ".xml";

}

try(XMLEncoder encoder = new XMLEncoder(new BufferedOutputStream(new FileOutputStream(filename)))){

encoder.writeObject(arr);

System.out.println("Serialization successful.");

}

catch(Exception ex){

System.out.println(ex.getMessage());

}

break;

case 3:

break;

default:

System.out.println("You've entered the wrong command.");

break;

}

break;

case 6:

System.out.println("\nChoose the method");

System.out.println("1. Standard deserialization");

System.out.println("2. XML deserialization");

System.out.println("3. Return");

System.out.println("=========================");

System.out.print("Your choise: ");

choise2 = scan.nextInt();

switch(choise2) {

case 1:

scan.nextLine();

System.out.print("Enter the name of file: ");

String filename = scan.nextLine();

if (filename.indexOf(".ser") == -1) {

filename += ".ser";

}

try(ObjectInputStream oos = new ObjectInputStream(new BufferedInputStream(new FileInputStream(filename)))){

arr.clear();

arr = (MyContainer<Event>) oos.readObject();

System.out.println("\nSerialization successful.");

}catch(Exception ex){

System.out.println(ex.getMessage());

}

break;

case 2:

scan.nextLine();

System.out.print("Enter the name of file: ");

filename = scan.nextLine();

if (filename.indexOf(".xml") == -1) {

filename += ".xml";

}

try(XMLDecoder decoder = new XMLDecoder(new BufferedInputStream(new FileInputStream(filename)))){

arr.clear();

arr = (MyContainer<Event>) decoder.readObject();

System.out.println("Serialization successful.\n");

}catch(IOException ex){

System.out.println( );

}

break;

case 3:

break;

default:

System.out.println("You've entered the wrong command.");

break;

}

break;

case 7:

System.out.println("\nChoose sorting field:");

System.out.println("1. Sort by event date");

System.out.println("2. Sort by event length");

System.out.println("3. Sort by number of people");

System.out.println("4. Return");

System.out.println("===========================");

System.out.print("Your choise: ");

choise2 = scan.nextInt();

switch(choise2) {

case 1:

arr.sort(new EventDateComparator());

System.out.println("\nData sorted\n");

break;

case 2:

arr.sort(new EventLengthComparator());

System.out.println("\nData sorted\n");

break;

case 3:

arr.sort(new EventPeopleNumberComparator());

System.out.println("\nData sorted\n");

break;

case 4:

System.out.println("\nReturning\n");

break;

default:

System.out.println("\nYou have entered the wrong number.\n");

break;

}

break;

case 8:

System.out.println("\nTerminating the program.");

stop = true;

break;

default:

System.out.println("You have entered the wrong number.");

break;

}

}while(!stop);

scan.close();

return arr;

}

private static Event inputNewEvent(){

Scanner scan = new Scanner(System.in);

int value;

boolean ready = false;

do {

System.out.print("\nEnter number of participants: ");

value = scan.nextInt();

if(value < 1)

{

System.out.println("Error. Wrong list size.\n");

}

ready = true;

}while(!ready);

ArrayList<String> list = new ArrayList<String>();

String temp;

System.out.println("Enter list of names:");

scan.nextLine();

for (int i = 0; i < value; i++) {

System.out.print(i+1 + ". ");

temp = scan.nextLine();

list.add(temp);

}

GregorianCalendar date = new GregorianCalendar();

System.out.print("Enter event year: ");

value = scan.nextInt();

date.set(Calendar.YEAR, value);

System.out.print("Enter event month: ");

value = scan.nextInt();

date.set(Calendar.MONTH, value-1);

System.out.print("Enter event day: ");

value = scan.nextInt();

date.set(Calendar.DAY\_OF\_MONTH, value);

System.out.print("Enter event hour: ");

value = scan.nextInt();

date.set(Calendar.HOUR\_OF\_DAY, value);

System.out.print("Enter event minute: ");

value = scan.nextInt();

date.set(Calendar.MINUTE, value);

date.set(Calendar.SECOND, 0);

System.out.print("Enter event address: ");

scan.nextLine();

temp = scan.nextLine();

System.out.print("Enter event description: ");

String description = scan.nextLine();

System.out.print("Enter event length: ");

value = scan.nextInt();

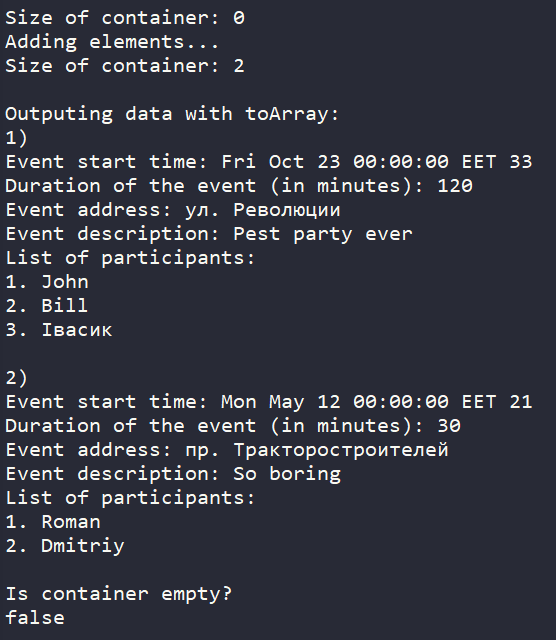
System.out.println("\nEvent added.\n");

Event newEvent = new Event(date,value,temp,list,description);

return newEvent;

}

}

**Результат роботи програми**

**Висновки**

При виконанні даної лабораторної роботи було набуто практичного досвіду з обробки параметризованих контейнерів.

Програма протестована, виконується без помилок.